Office de la propriété intellectuelle du Canada

Un organisme d'Industrie Canada

Canadian Intellectual Property Office

An Agency of Industry Canada

Bureau canadien des brevets

Certification

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

Canadian Patent Office

Certification

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,423,618,** on March 28, 2003, by **DIGITAL ACCELERATOR CORPORATION,** assignee of Yi Xiong, Mark Sauer and Meng Wang, for "Overcomplete Basis Transform-Based Motion Residual Frame Coding Method and Apparatus for Video Compression Systems".

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Agent certificateur/Certifying Officer

April 8, 2004

Date

Canada

(CIPO 68)
04-09-02

OPIC CIPO

PATENT APPLICATION:

OVERCOMPLETE BASIS TRANSFORM-BASED MOTION RESIDUAL FRAME CODING METHOD and APPARATUS FOR VIDEO COMPRESSION SYSTEMS

Inventor: Yi Xiong, Mark Sauer and Meng Wang
Assignee: Digital Accelerator Corporation, Vancouver, BC, Canada

References Cited

U.S. PATENT DOCUMENTS

5699121	12/1997	Zakhor and Neff 348/417; 348/422

OTHER REFERENCES

S. Mallat and Z. Zhang, "Matching Pursuits With Time-Frequency Dictionaries", IEEE Transaction in Signal Processing, vol. 41, No. 12, Dec. 1993.
R. Neff and A. Zakhor, "Very Low Bit Rate video Coding Based on Matching Pursuits," IEEE Transactions on Circuits and Systems for Video Technology, Feb. 1997, 158-171.
C. De Vleeschouwer and B. Macq, "New dictionaries for matching pursuit video coding", Proc. of the 1998 International Conference on Image Processing, vol. 1, 764-768.

ABSTRACT

This invention provides a method to compress digital moving pictures or video signals based on an overcomplete basis transform using a modified Matching Pursuit algorithm. More particularly, this invention focuses on the efficient coding of the motion residual picture, which is generated by the process of motion estimation and compensation. A residual energy segmentation algorithm (RESA) is proposed to obtain an initial estimate of the shape and position of high-energy regions in the residual. A progressive elimination algorithm (PEA) is proposed to reduce the number of matching evaluations in the matching pursuits process. RESA and PEA speed up the encoder by many times for finding the matched basis from the pre-specified overcomplete basis dictionary. Three parameters of the matched pattern form an atom, which defines index (into the dictionary) and position of selected basis, as well as the inner product between the chosen basis pattern and the residual signal. This invention proposes a novel atom position coding method using quad tree like techniques and a novel atom modulus quantization scheme. A simple and efficient adaptive mechanism is introduced for the quantization and position coding design to allow the invented system to operate properly in low, medium and high bit rate situations. These new algorithm components result in a much faster encoding process and better compression performance over previous matching pursuit based video coders.

BACKGROUND OF THE INVENTION

A sequence of pictures can occupy a vast amount of storage space and require very high transmission bandwidth when represented in an uncompressed digital form. Point to point digital video communication became practicable several years ago following advances in computer networks and signal compression technology.

The standardization effort for digital video compression was initiated in 1988. Currently, the Moving Picture Experts Group (MPEG) committee under ISO/IEC has completed both MPEG-1 and MPEG-2 standards; the MPEG-4 standard has also been completed, but new proposals are still being accepted. CCITT developed a series of recommendations – H.261, H.263 and H.263+ – that focus on low bit rate applications. All of these attempts at standardization utilize a two-step procedure to compress a video sequence. The first step uses a motion estimation and compensation algorithm to create a predicted video frame for the current video frame using the previous video frame – the difference between the current video frame and the predicted video frame is computed and is called the motion residual picture (MRP). The second step in the standard procedure is to code the MRP using the Discrete Cosine Transform (DCT). Such DCT-based systems do not perform well in all circumstances. At the low bit rates needed for personal video communication, DCT-based systems cause noticeable distortion and visible block artifacts. For high visual quality applications, such as DVD, the compression ratio achieved is quite low.

Motion residual pictures can be coded using other transform-based techniques. For example, discrete wavelet transforms (DWT) and overcomplete basis transforms can also be used. Zakhor and Neff presented a motion residual coding system in U.S. Patent 5699121 based on an overcomplete basis transform algorithm called matching pursuit; this was first proposed by Mallat and Zhang in *IEEE Transaction in Signal Processing*, vol. 41, No. 12, Dec. 1993. Zakhor and Neff's video coder improves both the visual quality and the PNSR over standard DCT-based video coders. However, their system is very slow and the compression performance is not optimized due to an ad-hoc design for matched basis position coding and quantization of the transform coefficients. Thus it is valuable to improve the overcomplete transform based video coding technique in both speed and efficiency.

SUMMARY OF THE INVENTION

The current invention is an improved coder for overcomplete-transform based residual picture coding, used for motion compensated video compression systems. This invention is analogous to previous matching pursuit video coders in that they decompose the residual image into a list of atoms, which represent basis functions from an overcomplete dictionary. However, the atom finding process is accelerated using the Residual Energy Segmentation Algorithm (RESA) and the Progressive Elimination Algorithm (PEA). The basis dictionary is often very large in order to characterize the features appearing frequently in motion residual images. To find an atom, prior matching pursuit methods have performed an exhaustive search of the basis dictionary; this exhaustive search is a very slow process. Instead of trying each basis in the dictionary, RESA identifies the approximate shape and position of regions with high energy in the motion residual images such that a good match can be found by comparison with a smaller subset of bases within the dictionary. Furthermore, PEA progressively removes pattern candidates from consideration by pre-computing the energy of search windows – significantly reducing the computation time needed to find the best match. Whenever a matched atom is found, the residual is updated by removing the part characterized by the atom. The foregoing steps of finding atoms and updating residuals are repeated until the desired compression bit rate or quality has been achieved.

The invention introduces a new modulus quantization scheme for matching pursuit with an overcomplete basis, which changes the atom finding procedure. The coefficients produced directly from the transform are continuous floating-point values, which require quantization for optimal digital coding under a bit budget. In the matching pursuit algorithm, it is necessary to use an in-loop quantizer – where each found atom is first quantized, and then used to update the residual image. Thus each atom affects the selection of subsequent atoms. If the quantizer is specified before coding begins, as in previous matching pursuit methods, it is difficult to optimize the quantization scheme as the optimal quantizer design depends on statistics of the list of chosen atom moduli. The disclosed quantization scheme chooses the quantizer adaptively during the atom searching process.

In addition to the atom modulus, the index of the chosen basis and the position of the atoms need to be transmitted in an overcomplete-transform based coder. The invention includes a method to code the atom position information efficiently. The atom position distribution forms a 2D map, where pixel values of one and zero represent the presence or lack thereof of atoms in each position respectively. A quadtree like technique has been developed to code the position map. The modulus and basis index information are embedded in the position coding. The atoms for different channels of color video (Y, U, V) are coded independently. Previous systems arrange all the atoms as a 1D list, and code the atom parameters one by one. It is hard to use the context information between atom positions and the special properties of atoms in different color channels for efficient coding by considering the list of all atoms as a 1D signal.

All atom parameters are transmitted after they have been encoded into a compressed version of the residual pictures. The associated decoder is also presented in the invention. The decoder reconstructs the residual picture through interpreting the coded bit stream back into atom parameters and combining the atom information to form the reconstructed stream of residual images that are then combined with the motion compensated image to form the reconstructed video stream.

In summary, the invention is a method for encoding motion residual pictures consisting of the steps of 1. forming the atom decomposition of the residual image in an overcomplete basis space using the modified matching pursuit algorithm, 2. choosing the modulus quantizer, 3. coding the atom position map, modulus as well as index for selected basis and 4. a method to decode residual signals that have been encoded using the method of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates the overall diagram of video compression systems that use the over-complete basis transform and associated coding methods of the invention.
FIG. 2 is an example of a motion residual picture processed by the invention.
FIG. 3 illustrates a simple dictionary with 16 bases.
FIG. 4 describes the whole atom decomposition process based on over-complete basis.
FIG. 5 describes the basic steps executed by residual energy segmentation algorithm (RESA).
FIG. 6 illustrates the first step of RESA.
FIG. 7 illustrates the second step of RESA: the horizontal growing scheme.
FIG. 8 illustrates the third step of RESA: the vertical growing scheme.
FIG. 9 describes the matching pursuit atom search using the progressive elimination algorithm (PEA).
FIG. 10 illustrates how to form the sub dictionary of matching basis and searching position candidates.
FIG. 11 illustrates the fast calculation of region energy.
FIG. 12 illustrates the parameters for one atom.
FIG. 13 is an example of an atom position map.
FIG. 14 is a flowchart illustrating the atom encoding process.
FIG. 15 is a flowchart illustrating the decoding of compressed residual signal.

Like reference numerals refer to corresponding parts throughout the several views of the drawings.

DETAILED DESCRIPTION OF THE INVENTION

FIG. 1 illustrates the associated processing executed by a video compression apparatus 10 that employs the invented residual picture coder 20. The video frame is initially processed by a motion estimator 30, which compares the current frame with one or two reference frames. In most cases, objects in the video change their position in successive frames while the background remains the same. Since the reference frames have been transmitted to video decoder 12, some regions in the reference frame can be used to construct the current frame. The motion estimator 30 identifies those regions within the reference frames that are similar to regions within the current frame. The motion compensator 32 produces the difference between those similar regions and combines them as a motion residual picture. The position relationships between similar regions are represented as motion vectors, which are processed by motion vector encoder 34. The atom decomposer 40 processes the residual picture first, and then the atom encoder 42 compresses the resulted atoms. The coded motion vectors and atoms are combined into one bit stream by the multiplexer 22. The compressed video is transmitted or stored by apparatus 24, which will deliver the video in compressed format to the video decoder 12.

The lower part of FIG. 1 illustrates the decoder 12, in which the demultiplexer 26 separates the compressed video signal, sending corresponding bits to the motion vector decoder 36 and the residual picture decoder 28 respectively. The motion reconstructor 38 forms the prediction frame from the reference frame and the motion vector. The invented residual decoder 28 reconstructs the residual. These two signals are added together to generate the final reconstructed video frame. FIG. 1 shows that residual coding is a core part of video compression.

FIG. 2 is an example motion residual picture for the Y color channel. The original residual has both negative and positive values. For proper displaying of the residual signal as a 256 level gray image, the pixel values in the residual are shifted and scaled so that pure gray means zero, while black and white represent negative and positive values respectively. Note that the residual picture contains several high-energy regions, which correspond to the motion of objects in the video.

Most signal compression techniques transform the original data into some more compact format through different kinds mathematical transformations. Some mathematical transforms, such as DCT and DWT, use a "complete basis" which forms an invertible transformation matrix. Recently, overcomplete basis and associated transformation algorithms have received considerable attention. The number of bases in an overcomplete basis dictionary is much larger than dimension of the original data. The benefit of an overcomplete basis is that the transformed coefficients are more effective in representing the true features in the original signal. There exist many mathematical methods to build a basis dictionary for different signals. Several dictionaries for video motion residual pictures have been designed and proved to cover features in residual pictures well. For example, a basis dictionary based on separable Gabor functions has been described by Neff and Zakhor in "Very Low Bit Rate Video Coding Based on Matching Pursuits", IEEE Transactions on Circuits and Systems for Video Technology, Feb. 1997, 158-171, and a basis dictionary based on Haar functions has been described by Vleeschouwer and Macq in "New dictionaries for matching pursuit video coding", Proc. of the 1998 International Conference on Image Processing, vol. 1, 764-768. FIG. 3 is a simple example dictionary containing 16 bases. Any of these dictionaries can be used with this invention. Focusing on the above-mentioned Gabor dictionary, there are 400 2D functions explicitly mentioned. However, it actually includes many more basis structures implicitly since each of those 400 2D functions can be placed at every possible position within the image. Using a frame size of 176X144 pixels implies that the dictionary actually contains 400X176X144=5.7 million bases structures – which makes it highly overcomplete. The transformation directly using the "matching pursuit algorithm" described by S. Mallat and Z. Zhang in "Matching Pursuits With Time-Frequency Dictionaries", IEEE Transaction in Signal Processing, vol. 41, No. 12, Dec. 1993, will take an extremely large number of computations to determine the transform coefficients. The matching pursuit for video compression, invented by Zakhor and Neff in US patent 5699121, reduces the calculation burden, however it remains computationally expensive. The invention focuses on the way to transform residual pictures based on general dictionaries, which is performed by the atom decomposer 40, and the way to code the transformed coefficients, which is the task of the atom encoder 42.

The operation of the atom decomposer 40 is fully described in FIG. 4. The first step (block 61) executed by the atom decomposer 40 is to find the initial search region. This step is realized by the residual energy segmentation algorithm (RESA) shown in FIG. 5. RESA is based on a general region growing idea. It initially selects a 2x2 block as start point for region growing (block 70). This step requires the division of the residual picture into 16x16 blocks, as shown in FIG. 6. The energy, which is the sum of the square of all pixel intensities, is computed for each block, and the block with highest energy is found, as block 71 shown in FIG. 6. Block 71 is further divided into four 8x8 sub-blocks, and the sub-block 72 with the highest energy is found. Within that 8x8 sub-block 72, the highest energy 2x2 block 73 is found – this block will be used as the start point for region growing.

The next step of RESA (block 74 in FIG. 5) is to check the 2x2 block in the left side of the current region. FIG. 7 illustrates this step of RESA. A threshold is calculated dynamically as

$$T = AE*\max(7-AU, 5)/10$$

where AU is the number of blocks that have been added on the left side of the start block, and AE is the average energy per 2x2 block of current region. If the energy of the checked 2x2 block is larger than the current threshold, the tested 2x2 block is grouped with the current region together to form a new larger current region. Otherwise, a stop point has been found on this side, and we do not group the blocks together. In a similar, symmetric fashion, check the 2x2 block on the right side of the current region. Continue growing left and right side iteratively, until stop points are found on both sides or the width of the rectangle has reached 32 (whichever comes first). A horizontal strip rectangle 75 is formed after this step, the dimension of the strip is 2*2m, 1<=m<=16.

The final step of RESA (block 76 in FIG. 5) is to grow the ,region vertically based on strip 75, as shown in FIG. 8. Assume the width of the strip 75 is W. Consider the 2*W strip rectangle above the current region, and a threshold

$$Ts= AEs*\max(7-AUs, 5)/10$$

Where AUs is the number of 2*W rectangles that have been added above the initial strip and AEs is the average energy per 2*W rectangle included in the current region. If the tested 2*W rectangle has an energy that is larger than a threshold, merge it into current region. Otherwise, a stop point has been found on this side. In a similar, symmetric fashion, check the 2*W rectangle below the current region. Continue growing up and down iteratively until stop points are found on both sides or the height of the current region has reached 32 (whichever comes first). In the end we get a rectangle 77 that has dimension 2n*2m, 1<=n,m<=16.

Returning now to FIG. 4, the process for finding the closest matched basis from the given dictionary is illustrated (block 62). The degree of matching between a basis and the residual picture is represented by the absolute value (modulus) of their inner product, which is called the atom modulus. A large modulus implies a good match. The process of determining this modulus requires computing a number of inner products, and selecting the one with the largest modulus as the current atom. This process is the slowest part in the matching pursuit algorithm. In the classical matching pursuit algorithm, the inner product between residual and each of the millions of elements in dictionary would need to be computed to determine the modulus. In the prior art, the 16*16 block with the highest energy in the residual picture is simply selected as the initial search region - each basis structure is centered at each location in the chosen block, and the inner product between the basis structure and the corresponding residual region will be computed. For a dictionary with 400 basis, this process requires 256x400=102400 inner product calculations. FIG. 9 illustrates the modified fast matching pursuit process.

The resulting RESA rectangle 77 in FIG. 8 provides an initial estimation for the shape of the high-energy feature. It is used to filter out bases in the dictionary that have a shape that is too different from the RESA rectangle. A subset of matching basis candidates (block 80) is then formed. Assume the width and height of rectangle 77 is w and h respectively. A sub-dictionary is formed containing all bases with shapes, specified by *width* and *height* respectively, that satisfies:

w-tw1<=width<=w+tw2 and h-th1<=height<=h+th2

where tw1,tw2,th1 and th2 are some values to confine the basis size. These values may be changed and adjusted according to the dictionary structure. The largest and smallest sizes of tested bases are illustrated as rectangle 90 and 91 in FIG. 10. Block B80 is a simple sub-dictionary example containing four bases.

RESA also estimates the location of high-energy features in the residual picture. The position candidates for matching bases are selected around the center of the RESA rectangle 77 (block 81). FIG. 10 shows a small rectangle 92 whose center is the same as RESA rectangle 77. It is supposed that all pixels within rectangle 92 will work as a center for the tested residual region. Rectangle 94 in FIG. 10 is an example whose center is point 93, or left-top corner of rectangle 92. The width (ws) and height (hs) of rectangle 92 is supposed to be variable with RESA rectangle 77. The relationship is

ws= 2*min(w/2 + 1,6) and hs=2*min(h/2+1,6)

The size of rectangle 92 can be decided by other rules or simply be fixed in an implementation. The basic idea is that a good match is located around the center of the RESA rectangle 77. Furthermore, any positions within rectangle 92 that already contain the center of an atom will not be considered for any new atoms. Point 95 in FIG. 10 is an example. The prior art does not place such a restriction. The idea is that if one atom provides a good fit, it should remove the energy around its center without introducing too much extra energy at its boundary. It makes no sense for the matching pursuit algorithm to return to the same position to produce a second atom. However, it does occur in practice because of bad pattern fitting and quantization. This restriction of forcing no position repetition has almost no effect on coding performance and makes the coding of the atom position information simpler.

The next processing step (block 89 in FIG. 9) is called the progressive elimination algorithm (PEA) for the residual matching pursuit. It is independent of the method used to form the testing basis sub-dictionary and set of testing positions. For example, PEA will still operate if the sub-dictionary is the entire dictionary, and set of position candidates is the set of coordinates comprising the whole residual picture. PEA is a method of finding the closest basis match more efficiently by progressively removing comparison candidates from consideration. This contrasts with classical matching pursuit, which compares all basis candidates at all possible positions. Initially the maximum modulus Mm is set to be zero (block 82). Next a basis b(k,l) is considered (block 83), where k and l represents the width and height of the 2D basis function. A same sized region centered at one position candidate r(k,l,p) in residual picture is formed (block 84). Block 85 compares ||r(k,l,p)||, the energy of r(k,l,p), with current maximum modulus (Mm) to decide if there is a need to calculate the inner product between r(k,l,p) and b(k,l). In order to explain this operation, recall the mathematical triangle inequality:

$$|<r(k,l,p),b(k,l)>| <= ||r(k,l,p)|| \, ||b(k,l)||$$

The objective of matching pursuit is to find the maximum |<r(k,l,p),b(k,l)>|. Assume current maximum modulus is Mm. For basis b(k,l) at position p, if corresponding residual r(k,l,p) satisfies ||r(k,l,p)|| ||b(k,l)||<=Mm, then

$$|<r(k,l,p),b(k,l)>| <=||r(k,l,p)|| \, ||b(k,l)|| <=Mm$$

Therefore, it is unnecessary to calculate inner product <r(k,l,p),b(k,l)> in this case, and the region r(k,l,p) is moved to next position. The norm of basis ||b(k,l)|| can be calculated a priori (actually most of basis are normalized, namely ||b(k,l)||=1), the only overhead for this test then is to calculate energy of r(k,l,p). Fortunately, there is a fast algorithm to get ||r(k,l,p)||, as described below.

Assume there are n different sizes of basis heights $\{v_1, v_2, \ldots, v_n\}$, and m different sizes of basis widths $\{h_1, h_2, \ldots, h_m\}$, that are increasingly ordered. The search rectangle dimension is hs*ws, and the left-top point of the search rectangle is p(x,y). The hs*ws*n*m energy values can be calculated through the following four steps:

Step 1: Calculate the energy for the $s=h_m+k$ columns (FIG. 11 shows an example of the columns). These columns are centered at $(x-h_m/2+i,y)$, $i=0,1,\ldots,s-1$. Their height is v_1. Their energy is represented as $C_{1,0}(0),C_{1,1}(0),\ldots C_{1,s}(0)$, and calculated as

$$C_{1,i}(0) = e(x-h_m/2+i,y-v_1/2) +\ldots+e(x-h_m/2+i,y)+\ldots+ e(x-h_m/2+i,y+v_1/2)$$

where e(x,y) represents the energy of pixels at position (x,y).

The energies for the next s columns with same coordinates as above strips and length v_2 can be computed as

$$C_{2,i}(0) = C_{1,i}(0) + \text{Extra } (v_2-v_1) \text{ Pixels Energy}, i=1,2,\ldots s$$

Generally, we have:

$$C_{j,i}(0) = C_{j-1,i}(0) + \text{Extra } (v_j-v_{(j-1)}) \text{ Pixels Energy}, i=1,2,\ldots s; j=1,2,\ldots n$$

Step 2: Calculate energy of columns that are vertical shift of columns in Step 1,

$$C_{j,i}(a)= C_{j,i}(a-1)- e(x-h_m/2+i,y-v_1/2+a-1) + e(x-h_m/2+i,y+v_1/2+a),a=1,\ldots,hs$$

where a represents the vertical shift number corresponding to y.

Step 3: Calculate the energies of regions with height vj, (j=1,...,n) and width $h_1, h_2, \ldots, h_m$ and center (x,y+a), (v=0,1, ..., hs) using

$$S_{j,1}(0,a)= C_{j,(hm-h1)/2}(a)+\ldots+C_{j,hm/2}(a)+\ldots+C_{j,(hm+h1)/2}(a)$$
$$S_{j,2}(0,a)= S_{j,1}(0,a)+ \text{Extra } (h_2 - h_1) \text{ columns' energy}$$

Generally,

$$S_{j,i}(0,a) = S_{j,i-1}(0,a) + \text{Extra } (h_i - h_{(i-1)}) \text{ columns' energy, } i=1,\ldots,m$$

Step 4: Calculate the energies of the first set of regions with vertical base length v_j, (j=1, ..., n) and horizontal base length h_i, (i=1,..,m) and center (x+b,y+a), (b=1,...,ws and a=1,...,hs) using

$$S_{j,i}(b,a) = S_{j,i}(b-1,a) - C_{j,(hm-hi)/2+b-1}(a) + C_{j,(hm+hi)/2+b}(a)\ .$$

The maximum modulus will be updated successively during the matching pursuit process; this will progressively confine the search space and hence, the name progressive elimination algorithm. Note that several bases can have the same sizes, thus one energy calculation may avoid several inner product calculations. The performance of PEA is also related with how fast a good match (not necessarily the best match) is found. Because large regions always contain more energy, it is supposed to test bases of larger dimension first.

If $\|r(k,l,p)\|>Mm$, block 86 is executed to calculate the inner product (p) between r(k, l, p) and b(k,l). Block 87 compares the absolute value of p with current maximum modulus Mm. If $|p|>Mm$, the new Mm is set as $|p|$ and the corresponding basis index and position are recorded. Regardless, we keep returning to block 84 until all search positions have been checked. Then blocks 83 through 88 are run repeatedly until all basis candidates have been tested. Finally, an atom is produced which includes three parameters: 1. The index of basis in the dictionary that gives the best match; 2. The location of the best match in residual picture with (x, y) coordinates; and 3. The inner product (p) between basis and residual picture. FIG. 12 shows an example of an atom on a residual image.

Returning to FIG. 4 again, the step after finding an atom is to record the atom parameters (block 63). Note in this stage, no quantization of the atom's modulus is performed. Decision block 64 will decide when to begin atom quantization. Its operation depends on the rate control goal defined by the video compression system. If the compression ratio is fixed, block 64 will check if bits are still available for more atoms. Because no actual coding has been done yet, the used bits for coding the current atoms has to be estimated. Let "Bip" represent the average bits for coding basis index and positions, "Bm(i)" represents the actual bits for the *i*'th atom's modulus without quantization. Allocating one bit for the sign of inner product (p), then the used bits for n atoms are estimated as

$$\text{Used Bits} = n*(Bip+1) + \sum(Bm(1) + Bm(2) + \ldots Bm(n))$$

where "Bip" is initialized according to experiential data for first residual frame, and set as real value of last frame. Bm(i) can be known exactly for each modulus. An important fact is that the modulus will be quantized later and will result in fewer bits to be used than currently estimated. Thus in this stage we will get fewer atoms than what we can code. If the video system wants to achieve a certain quality, which is defined by the mean square error (MSE) of the coded residual as compared to the actual residual, block 64 will compare the current MSE achieved with the MSE objective. The MSE after introducing one atom is updated according to follow equation:

$$MSE(n) = MSE(n-1) - p(n)*p(n)$$

where MSE(n) represents the MSE after using n atoms and p(n) represents the inner product of nth atom. Initially the MSE, or MSE(0), is set to the energy of original residual picture. After quantization is performed, MSE(n) will likely increase, and therefore no longer achieve the MSE objective. In summary, if bits are available or the quality goal has not been achieved, the residual picture will be updated based on the current atom (block 65), followed by a search for another atom recommencing at block 61. Otherwise if the bit or quality objective has been achieved; block 66 is executed for the quantization design. Residual updating, one step for the standard matching pursuit algorithm, can be described mathematically as

$$r(k,l,p) = r(k,l,p) - p(n)*b(k,l)$$

All regions not covered by current atom will be unchanged.

The design of the quantizer (block 66) is based on minimum modulus (Minm) value found so far. The quantization step size (QS) is set to

$$QS = \begin{cases} 32 & \text{if Minm>24;} \\ 16 & \text{if 2<Minm<=24;} \\ 8 & \text{if 6<Minm<=12} \\ 4 & \text{if Minm<=6;} \end{cases}$$

All atoms found up to this point, will be quantized using the above QS in the simple mid-read scalar quantization scheme. Next the residual picture is updated again according to the now quantized list of atom moduli 67. Assume that the atom coefficient before and after quantization are p(i), q(i) respectively (i=1,...,n). Assume that the corresponding bases are b(i), (i=1,...,n). The residual after *n* unquantized atoms is:

$$E(n) = (\text{Original Residual}) - p(1)b(1) - p(2)b(2) - \ldots.- p(n)b(n)$$

Its energy ||E(n)|| is known also. There are two ways to calculate the residual energy after quantization. The first way is to simply calculate the residual after quantization as

$$EQ(n) = (\text{Original Residual}) - q(1)b(1) - q(2)b(2) - \ldots. - q(n)b(n)$$

Another way is to update it recursively. Assume the quantization error for p(i) is Δp(i). Then the residual with only p(n) being quantized is

$$EQ(1) = E(n) - \Delta p(n)b(n) \text{ and } ||EQ(1)|| = ||E(n)|| + \Delta p(n)^* \Delta p(n) - 2\Delta p(n)<E(n), b(n)>$$

The residual with p(n),p(n-1) quantization becomes:

$$EQ(2) = EQ(1) - \Delta p(n-1)g(n-1)$$

This relationship is true recursively and can be written as

$$EQ(i) = EQ(i-1) - \Delta p(n-i+1)g(n-i+1), i=1,2,\ldots n, EQ(0) = E(n)$$

The corresponding energy is

$$|| EQ(i)|| =|| EQ(i-1)|| + \Delta p(n-i+1)\ \Delta p(n-i+1) - 2*\Delta p(n-i+1)< EQ(i-1),g(n-i+1)>$$

Finally, we will get EQ(n) and ||EQ(n)||, which is the start point for further atom finding. An important thing is that the list of atoms can be in any order for the recursive update to occur – the update does not need to occur in the order in which the atoms were found.

Because the moduli of atoms have been quantized, more atoms will now be necessary to achieve the rate control or quality objective. Therefore, block 68 is executed to find additional atoms. The process is the same as block 61 through 63. However, the atom moduli will be quantized immediately in this stage. We now need to deal with atoms whose moduli is smaller than (QS – QS/4), without throwing them out by setting their quantization value to zero. The scheme used is given below:

1. If the atom modulus is larger than (QS – QS/4) then quantizer it using QS;

2. Otherwise, if the atom modulus is larger than (QS/2-QS/8) then it is quantized as value QS/2;
3. Otherwise, if the atom modulus is larger than (QS/4-QS/16) then it is quantized as value QS/4;
4. Otherwise, if the atom modulus is larger than (QS/8-QS/32) then it is quantized as value QS/8.

Three levels down are good enough in practice although more levels can, in theory, be used.

After block 68, a real rate control logic unit is executed (block 69). Because the atoms are quantized in-loop in this stage, the achieved quality or actual number of bits used can be estimated. When the compression goal is achieved, the system will go into the atom encoder 42. Otherwise, the residual picture will be updated based on the quantized atom modulus and the system will return to block 68 to find next atom. For color video, a residual picture contains several channels, i.e. Y, U and V channels. The atom decomposer 40 will be used for each channel independently. With this scheme, each channel can have its own bit budget or desired quality goal. There are certain bit allocation methods, which can be used to allocate bit budgets for the different channels.

All the atoms are passed to the atom encoder 42 for output in a compressed form. This invention considers the atom distribution for each channel as a bi-value map, as illustrated in Fig. 13. The black pixels represent atoms in their corresponding position, while the white pixels represent a lack of atoms in that position. A quadtree-like technique is used to encode the positions containing atoms. The other parameters of each atom will bc encoded after the atoms position information, using variable length coding. The coding procedure for the atom parameter signal is illustrated in FIG. 14 and described in more detail below.

The first step of atom encoding is to decompose the whole atom map (Fig. 13) into n*n blocks (Block 101). The value n may either be 16 (for the Y channel) or 8 (for the U and V channels). For each n*n block, if there are no atoms in the block, a zero-bit is output; otherwise, a one-bit is output, and the block is processed further to locate the atoms to the decoder. A quadtree decomposition procedure is used for this, and is summarized in the following four steps:

Step 1. Initialize a list of atom blocks (LAB) with one element – the n*n block itself.

Step 2. Pick one element *e* from LAB. If *e*'s size is 1*1, output all atom parameters except for the position: namely the basis index, modulus and the sign of the inner product of *e* should be output, then proceed to step 4; otherwise, proceed to step 3.

Step 3. Output the atom pattern bits of the four sub-blocks of *e*: $a_1a_2a_3a_4$, where a_i (i=1,2,3,4) is one if there is an atom in the corresponding sub-block and zero otherwise. Put all sub-blocks *i* with a_i value equal to 1 into the cnd of the LAB and return to step 2.

Step 4. Check if the LAB is empty. If it is not empty, return to step 2; otherwise the encoding finishes for the one *n**n block.

The basis index and atom modulus should be coded using a variable length coder to conserve bits, since these signal parameters are not uniformly distributed. The atom position information is encoded implicitly by recording the decomposition procedure with the 0/1 bit data. It is important to use a variable length coding method to encode the atom pattern bits of the four sub-blocks: a1a2a3a4. There are 15 kinds of patterns for the atom pattern bits, a1a2a3a4 (note 0000 is impossible). However, some patterns, such as 1000, occur with much higher probability than other patterns. The probability of the different patterns can be estimated through experiments and used to create a variable length table design. Further, it should be noted that the probability distribution is variable for different channels and different atom densities. Therefore multiple tables are used, and the block's category information is encoded first so the decoder knows which table should be used for decoding purposes.

FIG. 15 illustrates the atom decoder 46, which performs operations that are the inverse of those performed by the atom encoder 42. First, atom decoder 46 receives one bit representing the status for the current n*n block. If the value is one, it is processed through the symmetric quadtree decomposition procedure. Initially, the n*n block is divided into four sub-blocks. The atom pattern bits for the four sub-blocks are decoded using the inverse variable length coding (VLC). Then all the sub-blocks with value 1 are put into list of atom blocks (LAB). The LAB is updated dynamically by decomposing each element in the LAB recursively and getting its atom pattern bits. If an element from the LAB is a 1*1 block, the atom basis index and the modulus should be decoded using the inverse VLC tables; the bit representing the sign of the inner product should then be read in. The atom decoder for one n*n block is finished if the LAB becomes empty.

The decoded atom parameter signal is then passed to the residual re-constructor 48, which forms the residual picture one channel by one channel using the method of classical matching pursuit. Initially the residual image is set to be 0's. Then each atom is added one by one using the following procedure. Let q(i) and b(i, k, l) represent the i'th atom coefficient and the corresponding 2D basis matrix respectively. If (x(i), y(i)) represents the location of the i'th atom, then the matrix q(i)*b(i, k, l) is added to the residual image constructed so far at position (x(i), y(i)) to get the new current residual. The process repeats until all atoms have been added for the channel. Once each channel has been decomposed, the process is finished and the residual image has been reconstructed.

Those familiar with the previous matching pursuit based video coding art will recognize a number of advantages associated with the disclosed techniques. The atom decomposition process based on an over-complete basis space has been speeded up significantly through a more accurate energy region estimation procedure and through the progressive candidate elimination algorithm. The atom modulus quantizer design is seamlessly chosen by the atom decomposition scheme, while the previous art specified the quantizer before the transformation began. Finally, the atom encoding process is more efficient because spatial relationships between the atoms are exploited by the invented quadtree-based decomposition scheme. The previous art collects all atoms into a 1D list, making it hard to efficiently code them.

DRAWING SHEETS

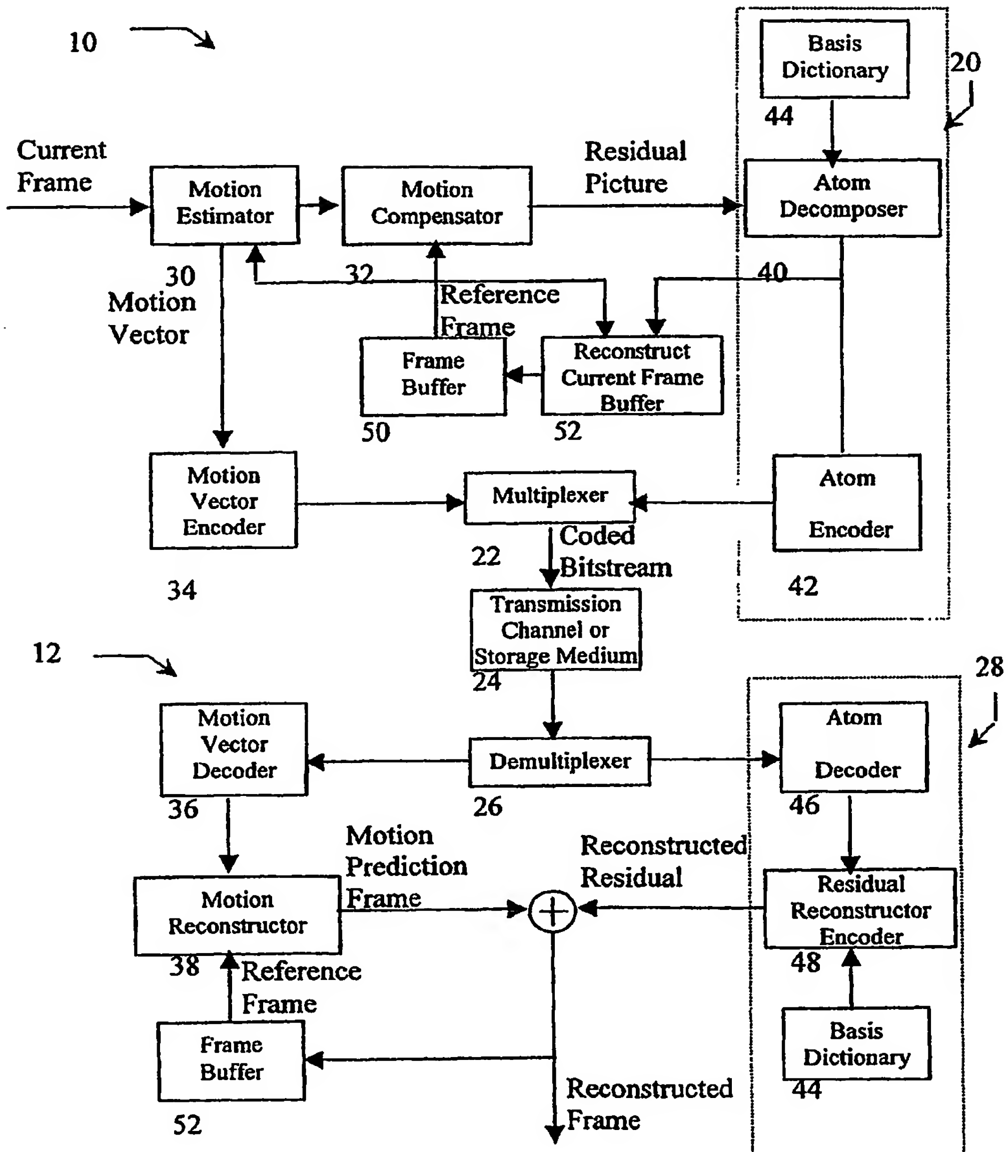


FIG. 1

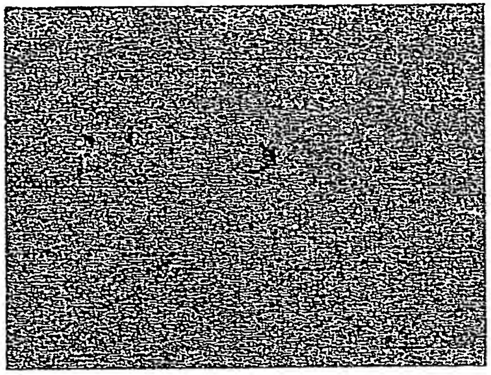


FIG. 2

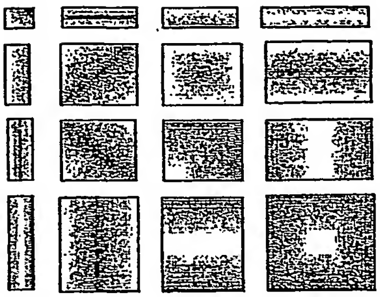


FIG. 3

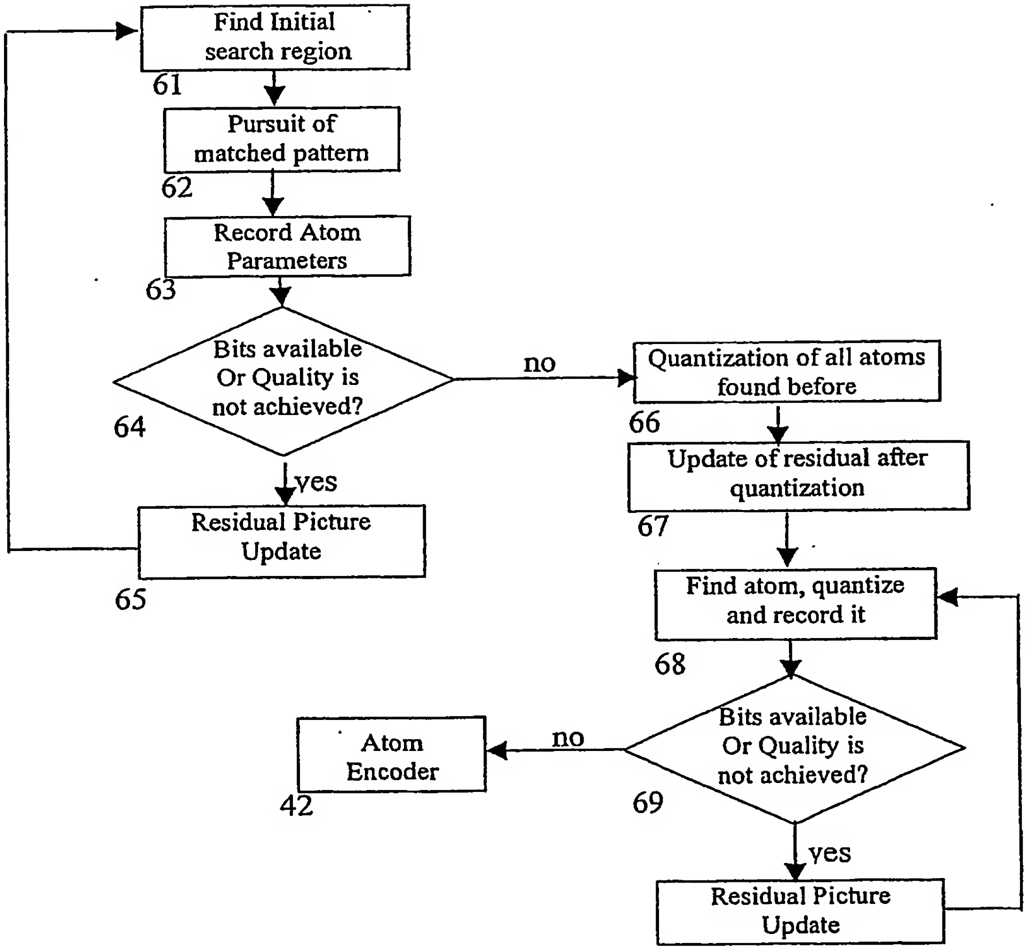

Find Initial search region
61
Pursuit of matched pattern
62
Record Atom Parameters
63
Bits available Or Quality is not achieved?
64
no
Quantization of all atoms found before
66
Update of residual after quantization
67
yes
Residual Picture Update
65
Find atom, quantize and record it
68
Atom Encoder
42
no
Bits available Or Quality is not achieved?
69
yes
Residual Picture Update

FIG. 4

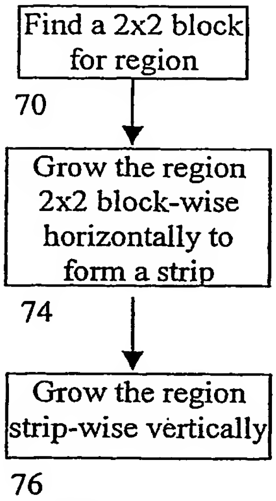

Find a 2x2 block for region
70
Grow the region 2x2 block-wise horizontally to form a strip
74
Grow the region strip-wise vertically
76

FIG. 5

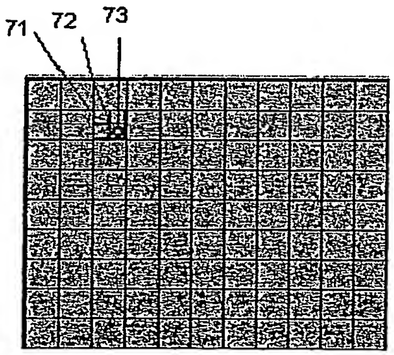


FIG. 6

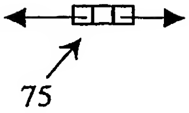


FIG. 7

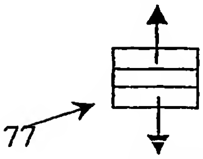
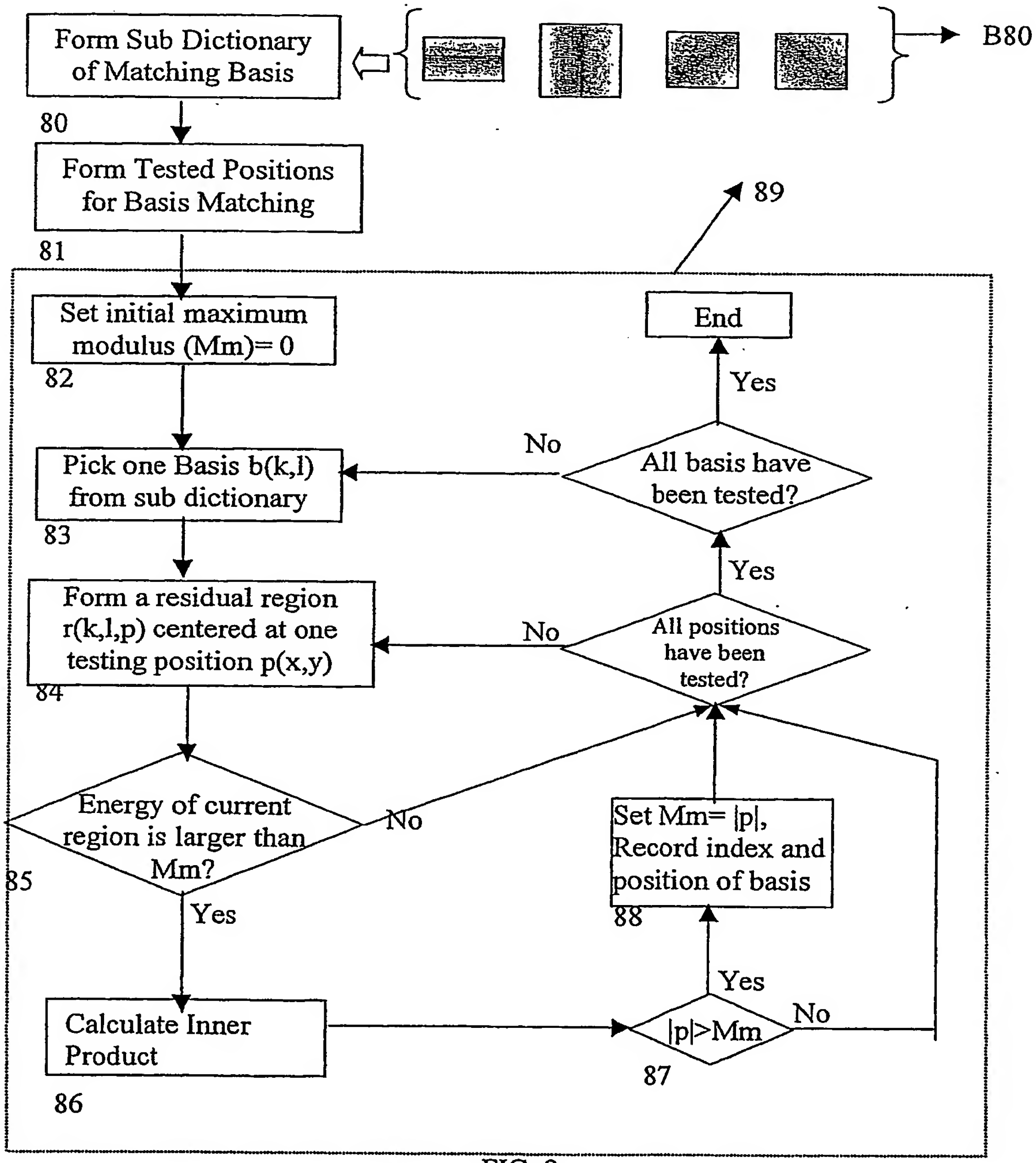


FIG. 8



Form Sub Dictionary of Matching Basis
B80
80
Form Tested Positions for Basis Matching
89
81
Set initial maximum modulus (Mm)= 0
End
82
Yes
No
Pick one Basis b(k,l) from sub dictionary
All basis have been tested?
83
Yes
Form a residual region r(k,l,p) centered at one testing position p(x,y)
No
All positions have been tested?
84
Energy of current region is larger than Mm?
No
Set Mm= |p|, Record index and position of basis
85
Yes
88
Yes
Calculate Inner Product
No
|p|>Mm
87
86

FIG. 9

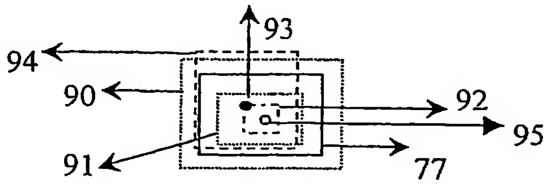


FIG. 10

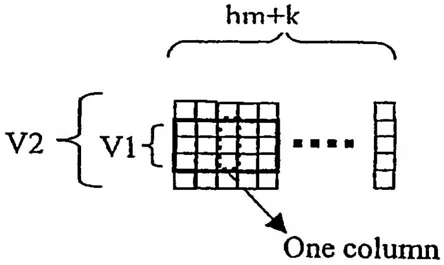


FIG. 11

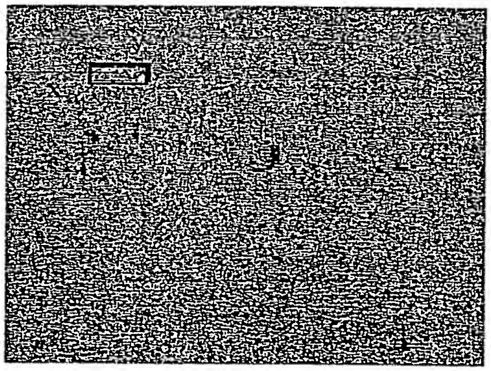


FIG. 12

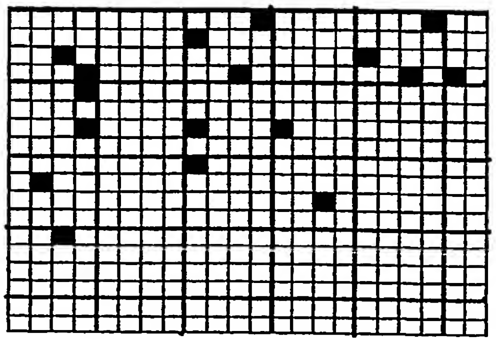


FIG. 13

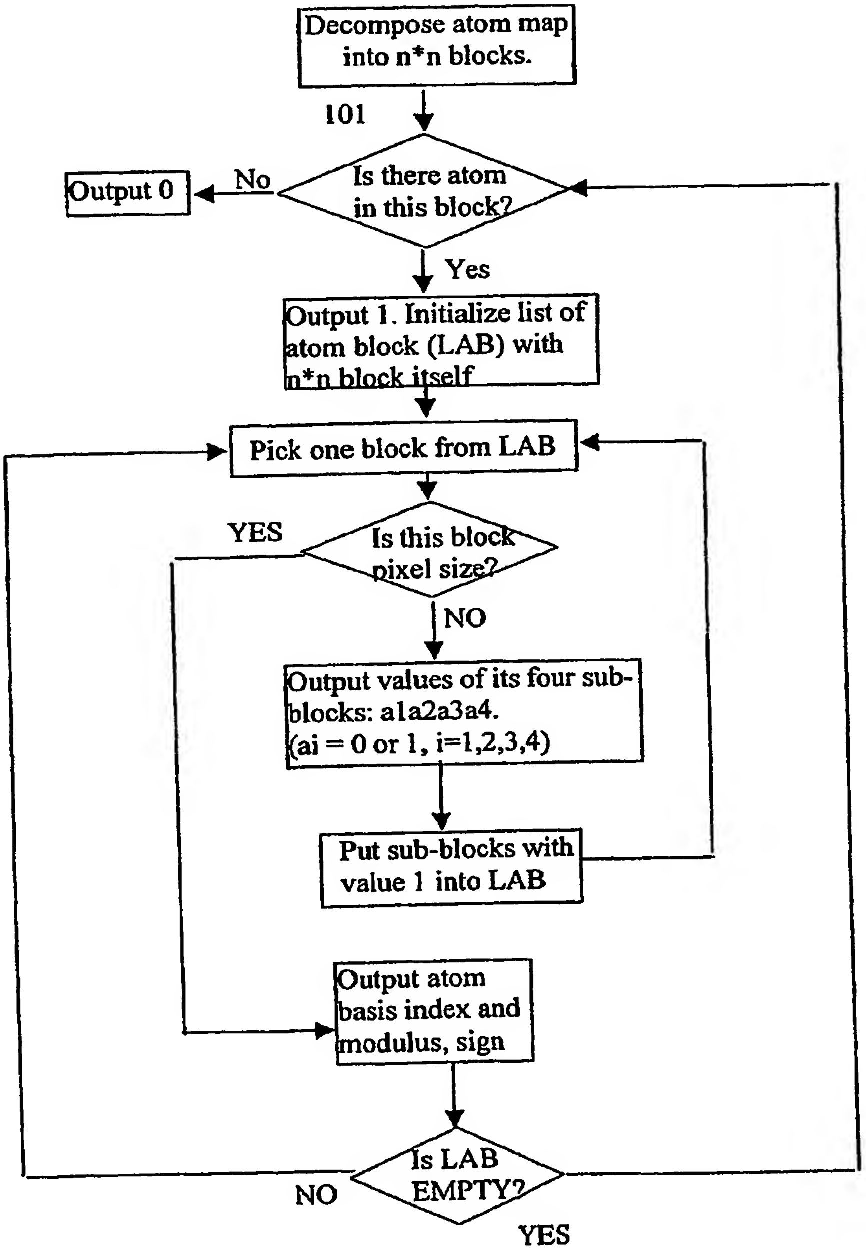

Decompose atom map into n*n blocks.
101
Is there atom in this block?
No
Output 0
Yes
Output 1. Initialize list of atom block (LAB) with n*n block itself
Pick one block from LAB
Is this block pixel size?
YES
NO
Output values of its four sub-blocks: a1a2a3a4. (ai = 0 or 1, i=1,2,3,4)
Put sub-blocks with value 1 into LAB
Output atom basis index and modulus, sign
Is LAB EMPTY?
NO
YES

FIG. 14

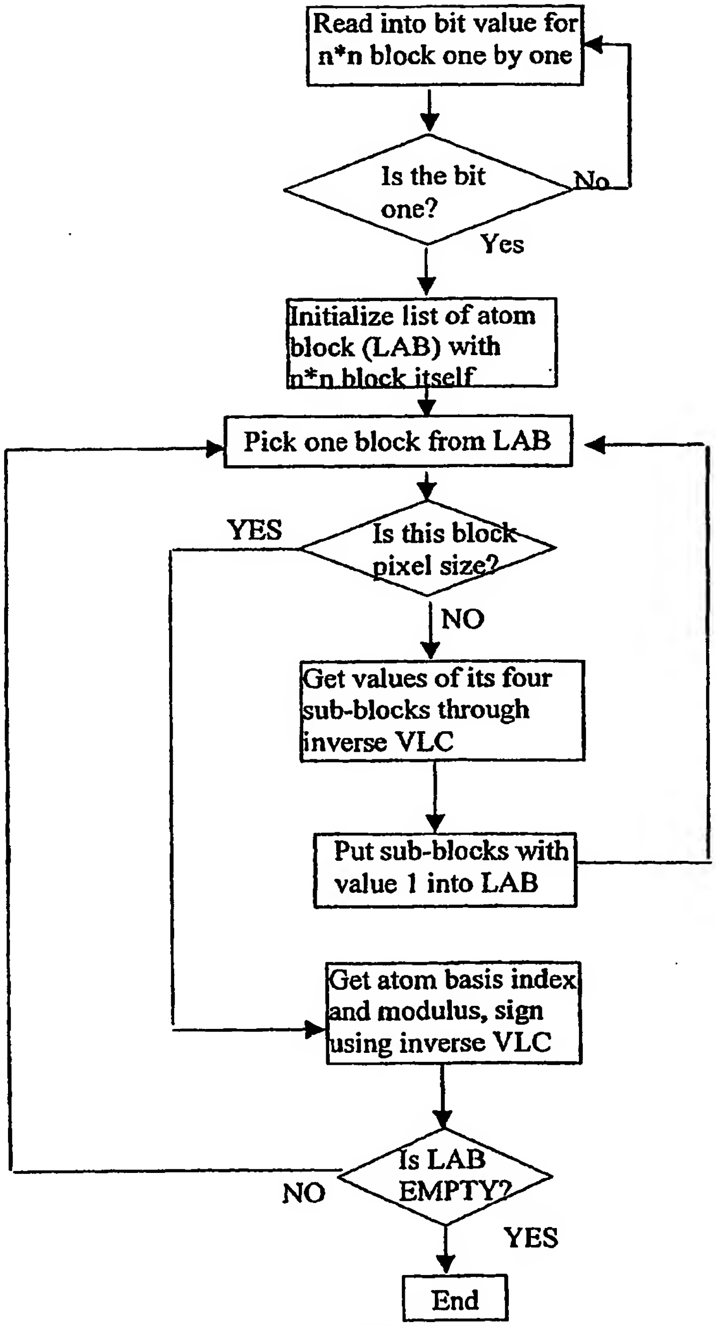

Read into bit value for n*n block one by one
Is the bit one?
No
Yes
Initialize list of atom block (LAB) with n*n block itself
Pick one block from LAB
YES
Is this block pixel size?
NO
Get values of its four sub-blocks through inverse VLC
Put sub-blocks with value 1 into LAB
Get atom basis index and modulus, sign using inverse VLC
NO
Is LAB EMPTY?
YES
End

FIG. 15